

3次元グラフィックスを用いた、ロボットの2足歩行シミュレーションに関する研究

愛知教育大学 教育学研究科
技術教育専攻 技術科内容学領域
梅田 寛康

1 はじめに

現在の産業分野では、2足歩行ロボットに代表される高度なロボット技術が発達し、研究、開発も著しく進歩している。ロボットは本来、複数の関節をもつ複数の部品からなるため、それらを動かすための運動方程式も複雑になる。そこで、我々の過去の研究[1]では、単純化した2足歩行モデル、すなわち、足が伸縮する倒立振り子モデル（質量は一点で、足裏は床と一点で接する）を使用し、歩行パターン生成をフリーソフトウェアの行列演算システムであるGNU Octaveによって行い、図示することで動作を確かめた。

本研究では、より現実的なモデルとして、関節（股関節・膝関節・足首関節）があり、足裏に一定の面積があるロボットをモデル化し歩行させることを目的とした。計算結果の図示は、3次元グラフィックスであるだけでなく、重力や物体の衝突など物理的に生じうる現象がコンピュータ画面上で表現可能な Open Dynamics Engine を採用した。

2 逆運動学

ロボットを構築するリンクの位置・姿勢と関節角度との関係を分析する理論は運動学と呼ばれる。運動学の中で、目標とする位置と姿勢が与えられたとき、それを実現する関節角度を求める方法が逆運動学である。逆運動学は数値解法と解析解法が存在し、本研究では解析解法を採用した。

解析解法では、解析対象となるロボットの機構の情報が必要となる。本研究は腰から下の下半身のみからなる2足歩行を対象とした。股関節にヨー角： θ_1 、ロール角： θ_2 、ピッチ角： θ_3 、の3自由度、膝関節にピッチ角： θ_4 、に1自由度、足首関節にロール角： θ_5 、

ピッチ角： θ_6 、の2自由度で、両脚で12自由度となっている。（図1）

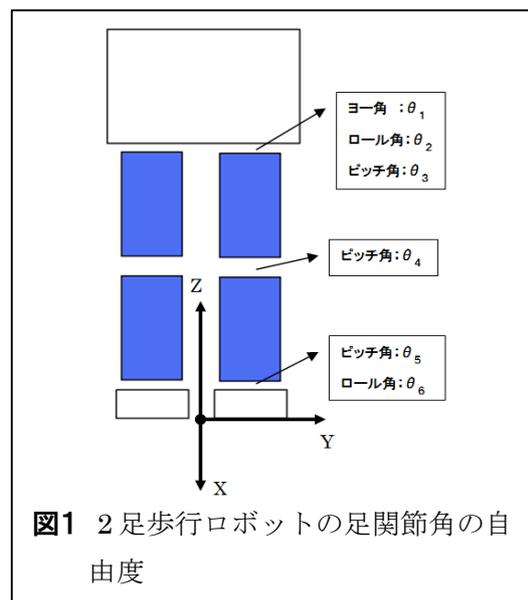
逆運動学計算による各関節角度は以下の式(1),(2),(3),(4),(5),(6)で求められる。

$$\theta_1 = \text{atan2}(b_x, b_y) \quad (1)$$

$$\theta_2 = -\text{atan2}(p_{3y}, -p_{3z}) \quad (2)$$

$$\theta_3 = \gamma - \text{atan2}\left(p_{3x}, \sqrt{p_{3y}^2 + p_{3z}^2}\right) \quad (3)$$

$$\theta_4 = \pi - \text{atan2}\left(\sqrt{1 - C_\beta^2}, C_\beta\right) \quad (4)$$



$$\theta_5 = \arctan2\left(p_{3x}, \sqrt{p_{3y}^2 + p_{3z}^2}\right) - \alpha \quad (5)$$

$$\theta_6 = \arctan2(p_{3y}, -p_{3z}) \quad (6)$$

3 Open Dynamics Engine (ODE)

Open Dynamics Engineはラッセル・スミス氏らが中心となって2001年から開発が進められている。オープンソースでありながら商用レベルの品質を備えている物理計算エンジンで、コンピュータゲーム、3Dの制作ソフトやシミュレータの物理計算エンジンとして広く使われている。[2]

ODEはC++言語に基づく3次元動力学演算フレームワークとして提供されており、フレームワークに含まれる関数を利用しロボットの物理計算やdrawstuffと呼ばれる3次元グラフィックスライブラリによって3次元描写を行う。ODEで作成した2足歩行ロボットの3次元描画を図2に示す。

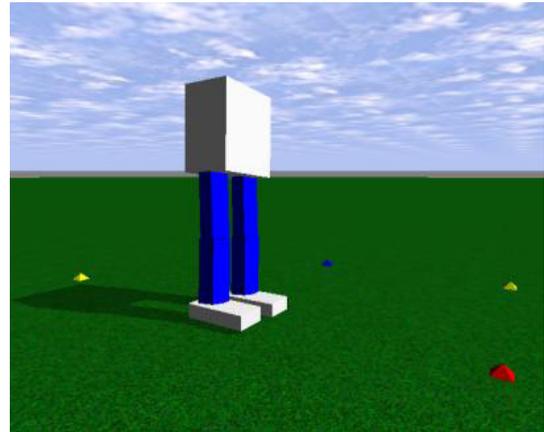


図2 ODEによる2足歩行ロボット

4 試行錯誤による歩行

2足歩行のアルゴリズムは文献[2]を参考にした。これは幾何学的に目標位置を順に指定するものである。アルゴリズムを下に示す

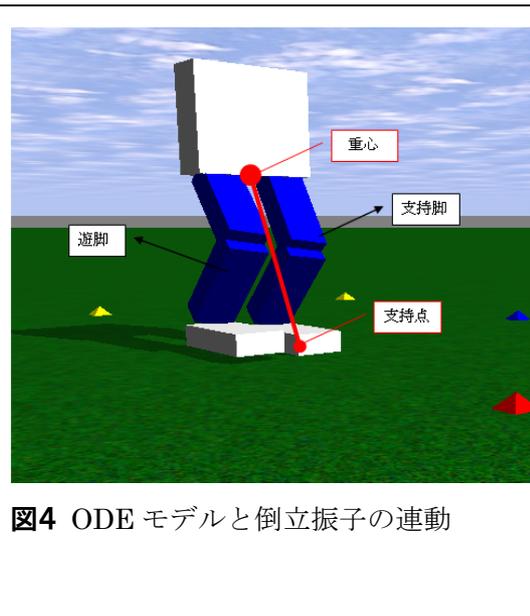
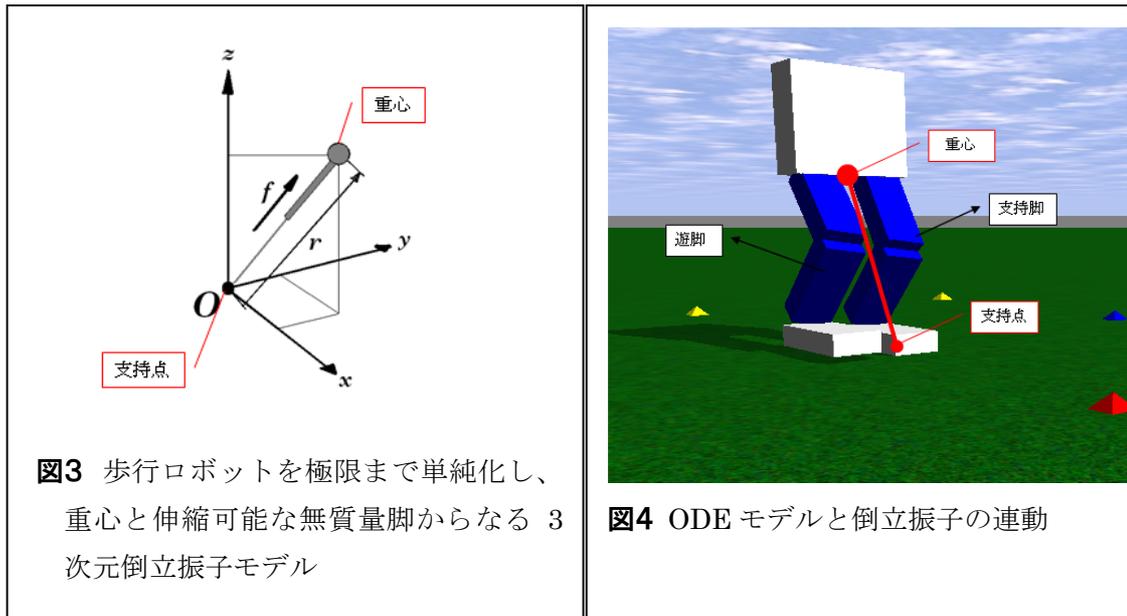
2足歩行アルゴリズム

1. **重心の移動**：遊脚となる脚を上げるために、もう一方の脚の制御を行い、重心を移動
2. **遊脚の離地**：遊脚となる脚をあげ、軌道上の最高点へ移動
3. **遊脚の着地**：遊脚を地面上の目標点へ移動し着地
4. **脚の切り替え**：重心を日幅の 1/2 前方へ移動し、次に遊脚となる脚を入れ替え、1. へ戻る

5 3次元倒立振り子モデルからより現実なモデルへ

図3に示す、3次元倒立振り子モデルを用いて実際のロボット運動を作る方法は、図4のように倒立振り子の重心とODEモデルの腰を連動させるものである。

そのために歩行開始時状態のODEモデルの重心位置、遊脚の軌道設定、腰（重心位置）と股関節の関係の3点を考え、腰と各脚の運動が決める。あとは逆運動学を用いて脚の各関節の運動を計算する。



6 考察

幾何学的に脚の目標位置を順に指定する方法で ODE によるロボットの歩行のシミュレーションを行うことはできた。しかし、試行錯誤的にパラメータを調整し歩行を実現する方法は効率が悪い。

過去の研究[1]では GnuOctave に微分方程式の計算を行う関数 `lsode` を用いて重心位置を計算した。C++にはそのような関数が存在しないのでオイラー法を用いて計算を行ったが、解が発散してしまった。これは誤差が累積し精度が落ちたためと考えられる。

C++で書かれた数値計算ライブラリである `liboctave` を用いることを考え、移植を行ったが時間が足らなかった。

ODE の 3 次元グラフィック描画機能によってとても美しいシミュレーションを行うことができ、ロボットの歩行運動を確認することができた。フレームワークに含まれる関数を用いることによって、簡単に物理計算を行うことができた。例えば、球、カプセル、円柱、直方体といった基本形状が決まっているものは、大きさや質量の調整も容易に行うことができ、重力計算や物体の形による衝突の計算も自動で行ってくれる。ジョイントにはモータが搭載されており、角度、角速度、トルクなどの制御を行うこともできるので、ロボットのシミュレーションにはとても有効的である。ODE はより現実に近い世界を容易につくることができ、ロボットの 2 足歩行の運動学の計算に集中することができた。

プログラミング言語が C++であるため、プログラミングの言語の勉強に時間がかかった。

7 関連研究

ODE はプログラミング言語に C++を使用したがる、その点でどうしても難しいと感じる。

LOGO などの小中学生向けの言語と、C++や Java などの高度なオブジェクト指向プログラミング言語との大きな隔たりを埋めるために、カーネギーメロン大学の研究者たちが開

発した Alice というプログラミング環境がある。[3] Alice は 3 次元モデルを使用してコンピュータアニメーションを作成するオブジェクト指向プログラミング言語である。Alice は教育向けのプログラミング言語である Squeak eToys に似ており、命令をドラッグアンドドロップできる特徴がある。Alice はユーザーが文法で混乱をせずに、プログラミングの理論を理解するのに役に立ち、オブジェクトなどのオブジェクト指向プログラミング言語特有の概念を集中して学ぶことができる。

Alice は 3 次元グラフィックス描画を用いるが ODE のような物理計算は行っていないため、現実世界の運動とは全く異なる。また、描画ライブラリがある程度決まっており、ODE のように物体と物体をつなげ制御する関節機能などもないため、自作のロボットやロボットの運動を描画しにくい。

描画という点ではタートルグラフィックスに注目したい。タートルグラフィックスは図形の描画を、画面上の絶対座標ではなく、タートルと呼ばれるカーソルを基準点とした相対座標で行うことができ、ユーザーは、自己と一体化しやすいタートルを操作してその軌跡を図形として描いたり色を塗ったりして楽しむ事が簡単に出来る。

しかし、3 次元タートルグラフィックスをサポートしている 3D-LOGO は 2 次元のタートルグラフィックスを 3 次元に拡張したものであり、線形 3 次元描画のため奥行など隠れて見えにくい部分が、理解しにくいという欠点もある。また、これを改良したという研究も特にない。[4]

言語の使いやすさ、描画の方法などの点を考え、一般ユーザー（エンドユーザー）であっても、2 足歩行ロボットの歩行制御のできるプログラミング言語の開発を行うことが今後の課題である。

8 おわりに

今後は、liboctave の移植を行い、ODE でのロボットの歩行を安定させる。そして、子供などのエンドユーザーでも抵抗なく 2 足歩行ロボットの歩行制御のできるプログラミング言語の開発を考えていきたい。

参考文献

- [1] 梅田寛康・鎌田敏之・加藤正義：「2足歩行ロボットのモデルベース歩行制御に関する研究」，第25回日本産業技術教育学会東海支部大会論文集, 2006.
- [2] 出村公成：「簡単！実装！ロボットシミュレーション」森北出版, 2007.
- [3] Charles W. Herbert: 「an introduction to programming using Alice」, Thomson Course Technology, 2006
- [4] 小谷善行・阿竹克人・高田正之・福田洋：「LOGO 空間プログラミング」，岩波書店, 1990